Hybrid Channel Access Scheduling in Ad Hoc Networks *

Lichun Bao
Computer Science Department
University of California, Santa Cruz
baolc@soe.ucsc.edu

J.J. Garcia-Luna-Aceves
Computer Engineering Department
University of California, Santa Cruz
jj@soe.ucsc.edu

Abstract

We present the hybrid activation multiple access (HAMA) protocol for ad hoc networks. Unlike previous channel access scheduling protocols that activate either nodes or links only, HAMA is a node-activation channel access protocol that also maximizes the chance of link activations using time- and code-division schemes. HAMA only requires identifiers of the neighbors within two hops from each node to schedule channel access. Using this neighborhood information, each node determines whether to transmit in the current time slot on a dynamically assigned spreading code. A neighbor protocol supplements HAMA with up-to-date two-hop neighborhood information by reliably propagating the one-hop neighbor updates through a novel random access technique. The throughput and delay characteristics of HAMA in randomly-generated multihop wireless networks are studied by analyses and simulations. The results of the analyses show that HAMA achieves higher channel utilization in ad hoc networks than a distributed scheduling scheme based on node activation, similar throughout as a well-known scheduling algorithm based on complete topology information, and much higher throughout than the ideal CSMA and CSMA/CA protocols.

1 Introduction

Channel access protocols for ad hoc networks can be contention-based or conflict-free. The contention-based approach started with ALOHA and CSMA [11] and continued with several collision avoidance schemes, of which the IEEE 801.11(b) standard for wireless LANs [5] being the most popular example to date. However, as the network load increases, network throughput drastically degrades because the probability of collisions rises, preventing stations from acquiring the channel.

*This work was supported by the U.S. Air Force/OSR under grant No. F49620-00-1-0330.

On the other hand, conflict-free access schemes schedule a set of timetables for individual nodes or links, such that the transmissions from the nodes or over the links are conflict-free in the code, time, frequency or space divisions of the channel. The schedules for conflict-free channel access can be established based on the topology of the network, or it can be topology independent.

Topology-dependent channel access control algorithms can establish transmission schedules by either dynamically exchanging and resolving time slot requests [4] [19], or prearrange a time-table for each node based on the network topologies. Setting up a conflict-free channel access timetable is typically treated as a node or link coloring problem on graphs representing the network topologies. The problem of optimally scheduling access to a common channel is one of the classic NP-hard problems in graph theory (k-colorability on nodes or edges) [6] [7] [15]. Polynomial algorithms are known to achieve suboptimal solutions using randomized approaches or heuristics based on such graph attributes as the degree of the nodes.

A unified framework for TDMA/FDMA/CDMA channel assignments, called UxDMA algorithm, was described by Ramanathan [14]. UxDMA summarizes the patterns of many other channel access scheduling algorithms in a single framework. These algorithms are represented by UxDMA with different parameters. The parameters in UxDMA are the constraints put on the graph entities (nodes or links) such that entities related by the constraints are colored differently. Based on the global topology, UxDMA computes the node or edge colorings, which correspond to channel assignments to these nodes or links in the time, frequency or code domain.

A number of topology-transparent scheduling methods have been proposed [3] [10] [12] to provide conflict-free channel access that is independent of the radio connectivity around any given node. The basic idea of the topology-transparent scheduling approach is for a node to transmit in a number of time slots in each time frame. The times when node i transmits in a frame corresponds to a unique code such that, for any given neighbor k of i, node i has

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2002**	2. REPORT TYPE	3. DATES COVERED **00-00-2002 to 00-00-2002**
4. TITLE AND SUBTITLE **Hybrid Channel Access Scheduling in Ad Hoc Networks**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**12**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

at least one transmission slot during which k and none of k's own neighbors are transmitting. Therefore, within any given time frame, any neighbor of i can receive at least one packet from i conflict-free. An enhanced topology transparent scheduling protocol, TSMA (Time Spread Multiple Access), was proposed by Krishnan and Sterbenz [12] to reliably transmit control messages with acknowledgments. However, TSMA performs worse than CSMA in terms of delay and throughput [12].

We propose a new hybrid activation multiple access protocol (HAMA), which supports broadcast, multicast and unicast communications through code- and time-division multiple access in wireless networks. HAMA requires only neighbor information within two hops from each node for computing channel access schedules, which relieves the dependence on global topology information, or network size and maximum degree, which UxDMA and TSMA have, respectively, and constitutes the minimum topology information needed to derive topology-dependent transmission schedules. Based on the two-hop neighbor information, each node computes the priorities of all the nodes in its two-hop neighborhood during each time slot to determine whether the node should transmit or receive in point-to-point or point-to-multipoint communications. The difference with the node-activation and link-activation protocols proposed in [1] is that HAMA schedules the channel access for broadcast while maximizing the chances of unicast at the same time, whereas the previous protocols are capable of supporting only node- or link-activation, but not both. Section 2 provides the basic assumptions on the behavior of the channel access and radio communication, and describes HAMA assuming that nodes have accurate knowledge regarding their two-hop neighborhood.

Section 3 presents a neighbor protocol for acquiring two-hop neighbor information and handling network dynamics. The protocol utilizes a similar approach to that of TSMA [3] using repetitive message transmissions. However, the neighbor protocol differs from TSMA in that it transmits short neighbor update messages using a random access scheme without acknowledgments, instead of transmitting regular data packets based on fixed codes with acknowledgments as in TSMA [12]. The minimum time required to deliver an update message by retransmissions with a given probability is derived as a function of the number of nodes in a neighborhood.

Section 4 compares the performance of HAMA through simulations and analyses with those of node-activation scheduling, idealized versions of CSMA and CSMA/CA, and UxDMA, which is a well-know scheduling algorithm based on complete topology information.

2 HAMA

2.1 Modeling of Network and Contention

We assume that each node is assigned a unique identifier, and is mounted with an omni-directional radio transceiver that is capable of communicating using DSSS (direct sequence spread spectrum) on a pool of well-chosen spreading codes. The radio of each node only works in half-duplex mode, i.e., either transmit or receive data packet at a time, but not both.

In multihop wireless networks, signal collisions may be avoided if the received radio signals are spread over different codes or scattered onto different frequency bands. Because the same codes on certain different frequency bands can be equivalently considered to be on different codes, we only consider channel access based on a code division multiple access scheme.

Time is synchronized at each node, and nodes access the channel based on slotted time boundaries. Each time slot is long enough to transmit a complete data packet, and is numbered relative to a consensus starting point. Although global time synchronization is desirable, only limited-scope synchronization is necessary for scheduling conflict-free channel access in multihop ad hoc networks, as long as the consecutive transmissions in any part of the network do not overlap across time slot boundaries. Time synchronization has to depend on physical-layer timing and labeling for accuracy, and is outside the scope of this paper.

The topology of a packet radio network is represented by an undirected graph $G = (V, E)$, where V is the set of network nodes, and E is the set of links between nodes. The existence of a link $(u, v) \in E$ implies that $(v, u) \in E$, and that node u and v are within the transmission range of each other, so that they can exchange packets via the wireless channel. In this case, u and v are called *one-hop neighbors* of each other. The set of one-hop neighbors of a node i is denoted by N_i^1. Two nodes are called *two-hop neighbors* of each other if they are not adjacent, but have at least one common one-hop neighbor. The neighbor information of node i refers to the union of the one-hop neighbors of i itself and the one-hop neighbors of i's one-hop neighbors, which equals

$$N_i^1 \cup (\bigcup_{j \in N_i^1} N_j^1) .$$

To ensure conflict-free transmissions, it is sufficient for nodes within *two hops* to not transmit on the same time, code and frequency coordinates [16]. Therefore, a node should at least know the topology information within two hops for conflict-free channel access scheduling. The operation of HAMA assumes that each node already knows its neighbor information within two hops. Section 3 describes how this information can be acquired.

2.2 Code Assignment

HAMA is a time-slotted code division multiple access scheme based on direct sequence spread spectrum (DSSS) transmission techniques. In DSSS, code assignments are categorized into transmitter-oriented, receiver-oriented or a per-link-oriented code assignment schemes (also known as TOCA, ROCA and POCA, respectively) in ad hoc networks (e.g. [9] [13]). HAMA adopts transmitter-oriented code assignment because of its broadcast capability.

We assume that a pool of well-chosen orthogonal pseudo-noise codes, $C_{pn} = \{c_k \mid k = 0,\ 1, \cdots\}$, is available in the signal spreading function. During each time slot t, a spreading code is assigned to node i, denoted by $i.\texttt{TxCode}$, as given by Eq. (1).

$$i.\texttt{TxCode} = c_k,\ k = \texttt{Hash}(i \oplus t) \ \mathbf{mod}\ |C_{pn}| \ . \qquad (1)$$

$\texttt{Hash}(x)$ is a fast message digest generator that returns a random integer by hashing the input value x. The sign '$\oplus$' is designated to carry out the concatenation operation on its two operands.

2.3 Nodal States

A node can be be a transmitter or a receiver for broadcast or unicast transmission during any given time slot. Broadcast transmissions take place over the channel encoded by the sender's spreading code, and none of the one- or two-hop neighbors of the sender is allowed to transmit. Multiple concurrent unicast transmissions can take place on different code-channels.

To allow each node to determine its state in any given time slot using only its two-hop neighborhood information, a dynamic priority is assigned to the node in the time slot. During a time slot t, the priority of node $i \in V$ is computed according to Eq. (2) using the same function $\texttt{Hash()}$ of Eq. (1).

$$i.\texttt{prio} = \texttt{Hash}(i \oplus t) \ . \qquad (2)$$

A node then derives its state by comparing its own priority with the priorities of its neighbors. We require that only nodes with higher priorities transmit to those with lower priorities. Accordingly, HAMA defines the following node states:

- R *Receiver*: The node has an intermediate priority among its one-hop neighbors.
- D *Drain*: The node has the lowest priority among its one-hop neighbors, and can only receive a packet in the time slot.
- BT *Broadcast Transmitter*: the node has the highest priority within its two-hop neighborhood, and can broadcast to its one-hop neighbors.
- UT *Unicast Transmitter*: the node has the highest priority among its one-hop neighbors, but not among its two-hop neighbors. Therefore, the node can only transmit to a selected subset of its one-hop neighbors.
- DT *Drain Transmitter*: the node has the highest priority among the one-hop neighbors of a *Drain* neighbor.
- Y *Yield*: The node could have been in either UT- or DT-state, but chooses to abandon channel access because its transmission may incur unwanted collisions due to potential hidden sources from its two-hop neighbors.

If node i determines that it can transmit (i.e., it has the BT-, UT- or DT-state), node i has to select a set of one-hop neighbors that can receive its packets. For convenience, we denote the receiver set of node i by $i.\texttt{out}$, and the packet queues for the eligible receivers by $i.\texttt{Q}(i.\texttt{out})$.

If a node j happens to be in reception state (R or D), node j has to choose a neighbor, denoted by $j.\texttt{in}$, which has the highest priority among its one-hop neighbors, and listens on the transmission code assigned to $j.\texttt{in}$. The reception code of node j is denoted by $j.\texttt{RxCode}$, equal to the transmission code of $j.\texttt{in}$. The details of how a node determines when to listen or transmit and in which channel are described next.

2.4 Operation of HAMA

Because we have a limited number of pseudo-noise codes for assignment in HAMA, it is possible that multiple nodes within two hops of each other are in one of the transmitting states and are assigned the same code. Many code assignment algorithms have been proposed based on a k-coloring on the corresponding graphs using either distributed and centralized strategies [2] [8], which require considerable amount of control information in order to resolve conflicts. HAMA chooses to "randomly" assign a code to each node in each time slot, and to resolve possible collisions using the neighbor-aware contention resolution algorithm (NCR) proposed in [1].

According to the NCR algorithm, each entity among a group of contending entities knows its direct and indirect contenders to a shared resource. Contention to the shared resource is resolved in each context according to the priorities assigned to the entities based on the context number and their respective identifiers. The entities with the highest priorities among their contenders are elected to access the common resource without conflicts.

```
   HAMA(i, t)
   {
       /* Every node is initialized in Receiver state. */
1      i.state = R;
2      i.in = -1;
3      i.out = ∅;

       /* Priority and TxCode assignments. */
4      for (k ∈ N_i^1 ∪ (⋃_{j∈N_i^1} N_j^1)) {
5          k.prio = Hash(t ⊕ k);
6          n = k.prio mod |C_{pn}|;
7          k.TxCode = c_n;
8      }

       /* Find UT and Drain. */
9      for (∀j ∈ N_i^1 ∪ {i}) {
10         if (∀k ∈ N_j^1, j.prio > k.prio)
11             j.state = UT; /* May unicast. */
12         elseif (∀k ∈ N_j^1, j.prio < k.prio)
13             j.state = D; /* A Drain. */
14     }

       /* If i is UT, see further if i can become BT. */
15     if (i.state ≡ UT and
16         ∀k ∈ ⋃_{j∈N_i^1} N_j^1, k ≠ i, i.prio > k.prio)
17         i.state = BT;

       /* If i is Receiver, i may become DT. */
18     if (i.state ≡ R and
19         ∃j ∈ N_i^1, j.state ≡ D and
20         ∀k ∈ N_j^1, k ≠ i, i.prio > k.prio) {
21         i.state = DT;

           /* Check if i should listen instead. */
22         if (∃j ∈ N_i^1, j.state ≡ UT and
23             ∀k ∈ N_i^1, k ≠ j, j.prio > k.prio)
24             i.state = R; /* i has a UT neighbor j. */
25     }

       /* Find dests for Txs, and srcs for Rxs. */
26     switch (i.state) {
27         case BT:
28             i.out = {-1}; /* Broadcast. */
29         case UT:
30             for (j ∈ N_i^1)
31                 if (∀k ∈ N_j^1, k ≠ i, i.prio > k.prio)
32                     i.out = i.out ∪{j};
33         case DT:
34             for (j ∈ N_i^1)
35                 if (j.state ≡ D and ∀k ∈ N_j^1, k ≠ i, i.prio > k.prio)
36                     i.out = i.out ∪{j};
37         case D, R:
38             if (∃j ∈ N_i^1 and ∀k ∈ N_i^1, k ≠ j, j.prio > k.prio) {
39                 i.in = j;
40                 i.RxCode = j.TxCode;
41             }
42     }

       /* Hidden Terminal Avoidance. */
43     if (i.state ∈ { UT, DT } and ∃j ∈ N_i^1, j.state ≠ UT and
44         ∃k ∈ N_j^1, k.prio > i.prio and k.TxCode ≡ i.TxCode)
45         i.state = Y;

       /* Ready to communicate. */
46     switch (i.state) { /* FIFO */
47         case BT:
48             if (i.Q(i.out) ≠ ∅)
49                 pkt = The earliest packet in i.Q(i.out);
50             else
51                 pkt = The earliest packet in i.Q(N_i^1);
52             Transmit pkt on code i.TxCode;
53         case UT, DT:
54             pkt = The earliest packet in i.Q(i.out);
55             Transmit pkt on code i.TxCode;
56         case D, R:
57             Receive pkt on code i.RxCode;
58     }
   } /* End of HAMA. */
```

Figure 1. HAMA Specification.

Similarly, provided that each node obtains an accurate knowledge of its neighbors within two hops, HAMA decides whether a node i transmits or receives a packet on an assigned code in time slot t (the context number). Figure 1 presents the specification of HAMA.

Lines 1-8 in Figure 1 compute the priorities and code assignments of the nodes within the two-hop neighborhood of node i using Eq. (2) and Eq. (1), respectively. Depending on the one-hop neighbor information of each neighbor $j \in N_i^1$ and itself, node i classifies the status of node j and itself into receiver (R or D) or transmitter (UT) state (lines 9-14).

If node i happens to be in UT-state, then node i further checks whether it can broadcast by comparing its priority with the priorities of its two-hop neighbors. If node i is in *Receiver* state, it checks whether it has a *Drain* state neighbor j to which it can transmit. If it has such a neighbor, before node i gets into DT-state, it needs to make sure that no other one-hop neighbor is going to transmit to it (lines 15-25).

Node i decides its receiver set if it is in transmitter state, or its sources if in receiver state. A receiver i always listens to its one-hop neighbor with the highest priority by tuning its reception code into that neighbor's transmission code (lines 26-42).

If a transmitter i unicasts (UT- or DT-state), it must avoid the hidden terminal problem, in which case node i's one-hop receiver may be receiving from two transmitters on the same code (lines 43-45).

Finally, if node i has come into transmission state, it selects the earliest arrived packet (FIFO-strategy) to its receiver set $i.\texttt{out}$ and sends out the packet on its transmission code. Note that in case node i is in BT-state, it may choose to send a unicast packet if broadcast buffer is empty. Otherwise, if it is a receiver (lines 46-58), node i listens to the channel for any packet from its one-hop neighbor with the highest priority.

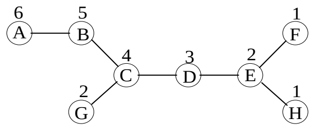

Figure 2. Example of HAMA Operation.

Figure 2 provides an example of how HAMA operates in a multihop network, assuming that each node has a different transmission code and that all nodes have accurate two-hop neighborhood information. In the figure, the priorities for the time slot are noted beside each node. Hence, from the standpoint of each node, node A has the highest priority among its two-hop neighbors, and becomes a broadcast transmitter. Nodes F, G and H are *Drain*-mode receivers, because they have the lowest priorities among their one-hop neighbors. Nodes C and E become transmitters to *Drain*s, because they have the highest priorities around their respective *Drain*s, and nodes B and D stay in *Receiver*-mode because of their low priorities. Notice that, in this example, only node A would be activated using a scheduling scheme that is purely based on node activation, because node C would defer to node A, and node E would defer to node C. This illustrates that hybrid channel access scheduling can provide substantial advantages over channel access based solely on node activation, which we confirm with our analyses in Section 4.

3 Neighbor Protocol

3.1 Random Access with Signals

In ad hoc networks, the two-hop neighbor information needed by topology-dependent scheduling protocols is acquired by each node propagating its one-hop neighbor states. However, exchanging neighborhood information among known and unknown neighbors cannot take advantage of the dynamic collision-free scheduling mechanisms described so far, because they assume *a priori* knowledge of the neighborhood. Hence, neighborhood information needs to be transmitted over a common channel on the best-effort basis. The neighbor protocol relies on an additional time section for coordinating neighbor information.

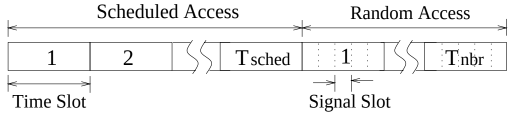

Figure 3. Time Division Scheme.

Figure 3 shows that the additional T_{nbr} time slots are is inserted after every T_{sched} scheduled-access time slots. In addition, every time slot for random access is subdivided into a number of smaller time segments, called *signal slots*, for transmitting short signals, each containing just a few hundred bytes.

Signals are used by the neighbor protocol for two purposes. One is for a node to say "hello" to its one-hop neighbors periodically in order to maintain connectivity. The other is to send neighbor updates when a neighbor is added, deleted or needs to be refreshed. In case of a new link being established, both ends of the link need to notify their one-hop neighbors of the new link, and exchange their complete one-hop neighbor information. In case of a link breaking down, a neighbor-delete update needs to be sent out. An existing neighbor connection also has to be refreshed periodically to the one-hop neighbors for robustness. If a neighbor-delete update is not delivered to some one-hop neighbors, those neighbors age out the obsolete link after a period of time.

3.2 Retransmission Scheduling

In order to keep inter-nodal connectivity current, each node broadcasts a signal packet on a common code-channel periodically. To avoid such periodic transmissions from synchronizing with one another, which would result in undue collisions of signal packets, the neighbor protocol adds random jitters to the interval value between signal packet transmissions.

However, because of the randomness of signal packet transmissions, it is still possible for a signal sent by a node to collide with signals sent by some of its two-hop neighbors. Due to the lack of acknowledgments in signal transmissions, multiple retransmissions are needed for a node to reassure the delivery of the same message to its one-hop neighbors, include "hello" and neighbor updates.

Without acknowledgments, retransmissions of a signal packet can only achieve a certain probability of delivery. Even though the message delivery probability approaches one as the neighbor protocol sends out the same message in signals repetitively for an extended period of time, the neighbor protocol has to regulate the rhythm of sending signals, so that the desired probability of the message delivery is achieved with a small minimum number of retransmissions in the shortest time, while incurring the least amount of interference to other neighbors' signal transmissions.

We analyze the time interval and the number of retransmissions needed to achieve a certain probability of message delivery by broadcasting signals.

For simplicity, denote the number of neighbors within two hops by N, the retransmission interval by T in terms of the number of signal slots, the number of retransmissions by

n, and the desired probability of message delivery by p. After a period during which the neighbor protocol operates, we assume that the signal slots chosen by two-hop neighbors to transmit signals are uniformly distributed over the interval T. Therefore, the probability of a successful transmission is $(1-1/T)^N$. When a single message is transmitted n times, the probability p of at least one successful delivery to all one-hop neighbors satisfies the following formula:

$$1-\left(1-\left(1-\frac{1}{T}\right)^N\right)^n = p$$

which gives

$$n = \frac{\ln(1-p)}{\ln\left(1-\left(1-\frac{1}{T}\right)^N\right)} \; . \qquad (3)$$

Hence, the duration of the required retransmissions is represented by function:

$$f(T) = T \cdot n = \frac{T\ln(1-p)}{\ln\left(1-\left(1-\frac{1}{T}\right)^N\right)} \; . \qquad (4)$$

Because a signal needs to be statistically delivered to one-hop neighbors as soon as possible, the parameter T in function should be chosen such that $f(T)$ is minimal over given N and p. Let $f'(T)=0$, we get

$$\frac{1}{\ln(1-\left(1-\frac{1}{T}\right)^N)} \cdot \frac{N(1-\frac{1}{T})^N}{1-\left(1-\frac{1}{T}\right)^N} \cdot \frac{1}{T-1} = -1 \; , \qquad (5)$$

which becomes independent of p.

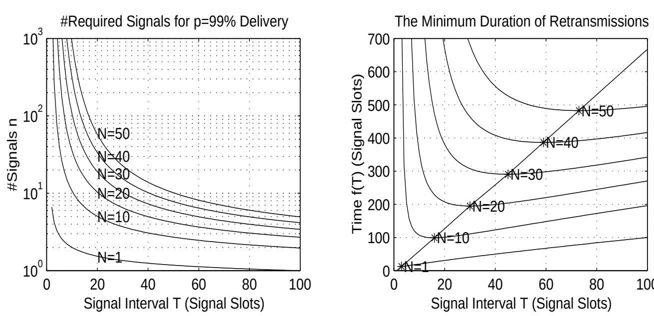

Figure 4. The Minimum Retransmissions and Time Required to Deliver Signals with $p = 0.99$

To find out the relation between T and N from Eq. (5), Eq. (3) and (4) are plotted in the left and right diagrams of Figure 4, respectively, when the required message delivery probability is $p = 0.99$. As shown in the figure, the minimum number and duration of retransmissions required to achieve the desired probability of message delivery are not constant, but varies depending on the signal retransmission interval T.

It is not hard to prove that Eq. (5) has only one root T in the form of N, which indicates that there is only one minimal point on each curve of the right diagram of Figure 4. Assume that N is large, and $T \approx kN$, Eq. (5) becomes

$$\frac{1}{\ln(1-e^{-1/k})} \cdot \frac{Ne^{-1/k}}{1-e^{-1/k}} \cdot \frac{1}{kN} + 1 \approx 0 \; ,$$

which can be solved using numeric estimation, and gives $k \approx 1.44$, meaning that, when the signal transmission interval is 1.44 times the number of neighbors within two hops, the time required to statistically deliver a signal to all one-hop neighbors becomes the shortest.

Applying $T \approx 1.44N$ ($N \gg 1$) to Eq. (3), n becomes:

$$n = \frac{\ln(1-p)}{\ln(1-\left(1-\frac{1}{1.44N}\right)^N)} \approx 1.45\ln\frac{1}{1-p} \; ,$$

which only depends on p. When $p = 0.99$, $n = 6.7$.

When N is small, a more detailed linear relation between T and N has to be considered, which is $T = 1.44N + 1.55$, derived from the minimum points in the right diagram of Figure 4.

Using the above results, for instance, if a node has $N = 20$ neighbors within two hops, then the signal packet interval is set to $T = 1.44N = 29$ signal slots, and the same message has to be retransmitted for $n = 7$ times to achieve 0.99 delivery rate. Accordingly, the duration of the retransmissions is $f(T) = nT \approx 194$ signal slots, matching the result in the right diagram of Figure 4.

The interval values have been based on signal slots. We compute the allocation of random-access time slots with regard to the absolute latency L needed by the neighbor protocol. As we stated in section 3.1, every T_{sched} time slots for scheduled access are followed by T_{nbr} time slots for random access to send signals. Therefore, the latency of delivering a message with the desired probability is related with three factors: (a) the duration of regular time slots and signal slots, (b) the portion of time for random access, and (c) the channel bandwidth. The duration of regular time slots and signal slots are determined by the bandwidth and the sizes of packets carried in these slots, and we denote the signal slot duration as t_s. Then, the portion of random access sections for achieving a desired latency L for message delivery satisfies:

$$\frac{T_{nbr}}{T_{nbr}+T_{sched}} = \frac{Tnt_s}{L} \; .$$

The more neighbors a node has, the longer the interval value T is set for signal retransmissions and the more the portion of time needed for random access. For instance, if the neighbor protocol is to handle up to a moderate number of neighbors within two hops, such as $N = 20$, the signal slot lasts $t_s = 1ms$, the message delivery desires probability $p = 0.99$ and latency is $L = 3s$, then the portion of time for random access should practically be set to

$$\frac{T_{nbr}}{T_{nbr} + T_{sched}} = \frac{1.44N \cdot 1.45 \ln \frac{1}{1-p} \cdot t_s}{L} = 6.4\%.$$

4 Performance

4.1 Channel Access Probability

In a fully connected network, it comes natural that the channel bandwidth is evenly shared among all nodes using HAMA, because the priorities of nodes are uniformly distributed. However, in a multihop ad hoc network scenario with random node placements, bandwidth allocation to a node is much more complex, and is a direct function of the channel access probability for the node. In the following, we derive the channel access probability for HAMA, and show that HAMA provides fair access to the channel and better bandwidth utilization than a dynamic scheduling protocol based on node activation only.

The randomly generated network topology is modeled as a result of independently and uniformly placing many nodes on an infinitely large two-dimensional area with node density ρ, where the probability of having k nodes in an area of size S follows Poisson distribution:

$$p(k, S) = \frac{(\rho S)^k}{k!} e^{-\rho S} .$$

The mean of the number of nodes in the area of size S is ρS.

Based on this modeling, the average channel access probability of each node, denoted by q_{HAMA}, is related with node density ρ and node transmission range r. q_{HAMA} includes the three mutually exclusive transmission probabilities of a node entering BT-, UT- or DT-state.

Let N_1 be the average number of one-hop neighbors covered by circular area under the radio transmission range of a node, we have $N_1 = \rho \pi r^2$.

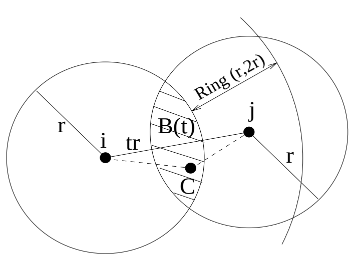

Figure 5. Becoming Two-Hop Neighbors

Let N_2 be the average number of one- and two-hop neighbors. As shown in Figure 5, two nodes become two-hop neighbors only if there is at least one common neighbor in the shaded area. The average number of nodes in the shaded area is:

$$B(t) = 2\rho r^2 a(t) ,$$

where

$$a(t) = \arccos \frac{t}{2} - \frac{t}{2}\sqrt{1 - \left(\frac{t}{2}\right)^2} . \quad (6)$$

Thus, the probability of having at least one node in the shaded area is $1 - e^{-B(t)}$ according to the Poisson distribution. Adding up all nodes covered by the ring $(r,\ 2r)$ around the node, multiplied by the corresponding probability of becoming two-hop neighbors, the average number of two-hop neighbors of a node is:

$$\rho \pi r^2 \int_1^2 2t \left[1 - e^{-B(t)}\right] dt ,$$

which results in:

$$N_2 = N_1 \left\{1 + \int_1^2 2t \left[1 - e^{-B(t)}\right] dt\right\} .$$

For convenience, symbol $T(N)$, $U(N)$ and $W(N)$ are introduced to denote three probabilities when the average number of contenders is N.

$T(N)$ denotes the probability of a node winning among its contenders. Because the number of contenders follows Poisson distribution with mean N, and that all nodes have equal chances of winning, the probability $T(N)$ is the average over all possible numbers of the contenders:

$$T(N) = \sum_{k=1}^{\infty} \frac{1}{k+1} \frac{N^k}{k!} e^{-N} = \frac{e^N - 1 - N}{N e^N} .$$

Note that k starts from 1 in the expression for $T(N)$, because a node with no contenders does not win at all.

$U(N)$ is the probability that a node has at least one contender, which is simply $1 - e^{-N}$. $W(N)$ is introduced to denote

$$W(N) = U(N) - T(N) .$$

Now we patch up the various notation introduced so far to derive the average probability of channel access in different transmissions states.

As N_2 denotes the average number of two-hop neighbors,

$$p_1 = T(N_2)$$

is the probability that i becomes BT.

The chances of unicast in either UT- or DT-state depends on the number of one-hop neighbors of the source and the

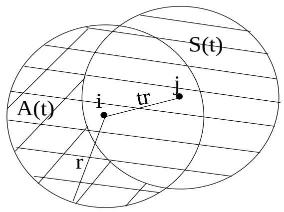

Figure 6. The Unicast Between Two Nodes

destination as well as the distance between them. For instance, when node i contends for transmission to node j, the combined one-hop coverage has to be considered as illustrated in Figure 6. Given that the transmission range is r, the average number of nodes within the combined coverage of the two circles, separated by tr ($0 < t < 1$), is

$$S(t) = 2\rho r^2 \left[\pi - a(t)\right] \ ,$$

where $a(t)$ is the same as in Eq. (6). The average number of nodes within the side lobe of two overlapping circles is

$$A(t) = 2\rho r^2 \left[\frac{\pi}{2} - a(t)\right] \ .$$

In general, if node i ever transmits in UT- or DT-state, then i must have at least one neighbor, in which case the probability is given by

$$p_2 = U(N_1) \ .$$

Then, if node i becomes a DT, node i needs to have a *Drain* neighbor j, and node i has the highest priority among node j's one-hop neighbors. In addition, there has to be at least another node in the left-side lobe of Figure 6 with higher priority than node i, preventing node i from becoming a BT or a UT. Thereof, the conditional probability of node i becoming DT is:

$$\sum_{k=1}^{\infty} \frac{A(t)^k}{k!} e^{-A(t)} \frac{k}{k+1} \cdot \sum_{k=0}^{\infty} \frac{N_1^k}{k!} e^{-N_1} \frac{1}{k+2} \frac{1}{k+1}$$
$$= \frac{T(N_1)}{N_1} W(A(t)) \ .$$

The probability density function (PDF) of node j at position t is $p(t) = 2t$. Therefore, integrating the above expression on t over range $(0, 1)$ with PDF $p(t) = 2t$ gives the average conditional probability of i becoming a DT, denoted by p_3:

$$p_3 = \frac{T(N_1)}{N_1} \int_0^1 2t \, W(A(t)) \, dt \ .$$

Alternatively, if node i becomes a UT with regard to node j, there has to be at least one higher-priority two-hop neighbor than node i, and node i has the highest priority among nodes covered in the combined region of Figure 6, which is interpreted into the following expressions evolved:

$$\sum_{k=1}^{\infty} \frac{[N_2 - S(t)]^k}{k!} e^{-(N_2 - S(t))} \frac{k}{k+1} \cdot \sum_{k=0}^{\infty} \frac{1}{k+2} \frac{S(t)^k}{k!} e^{-S(t)}$$
$$= \frac{W(N_2 - S(t)) \, W(S(t))}{S(t)} \ .$$

Using the PDF $p(t) = 2t$ for node j at position t, the integration of the above result on t over range $(0, 1)$ gives the conditional probability of i becoming a UT, denoted by p_4:

$$p_4 = \int_0^1 2t \, \frac{W(N_2 - S(t)) \, W(S(t))}{S(t)} \, dt \ .$$

In all, the average channel access probability of a node in the random ad hoc network is the chances of the three mutually exclusive events of becoming BT-, UT- or DT-state transmitters, which is given by

$$\begin{aligned} q_{HAMA} &= p_1 + p_2(p_3 + p_4) \\ &= T(N_2) + U(N_1) \cdot \left(\frac{T(N_1)}{N_1} \int_0^1 2t \, W(A(t)) \, dt \right. \\ &\quad \left. + \int_0^1 2t \, \frac{W(N_2 - S(t)) \, W(S(t))}{S(t)} \, dt \right) \ . \end{aligned} \tag{7}$$

The derivation of q_{HAMA} has made four simplifications. Firstly, we assumed that the number of two-hop neighbors also follows a Poisson distribution, just like that of one-hop neighbors. Secondly, we let $N_2 - S(t) \geq 0$ even though N_2 may be smaller than $S(t)$ when the transmission range r is small. Thirdly, only one neighbor j is considered when making node i become a DT or a UT, although node i may have multiple chances to do so owning to other one-hop neighbors. Lastly, we assumed that infinitely many codes are available such that hidden terminal collision on the same code was not considered. The results of the simulation experiments reported in Section 4.3 validate these approximations.

Using Eq. (7), the relation between transmission range and the *per-node* channel access probability in HAMA is shown in Figure 7, assuming a network density of $\rho = 0.0001$, which is equivalent to placing 100 nodes on a 1000×1000 square plane.

The figure also shows the channel access probability for a channel access protocol based solely on node activation, similar to the part of HAMA that selects BT-state transmitters. Accordingly, the first term in Eq. (7) provides the per-node channel access probability in the node activation scheme [1], which is

$$q_{na} = T(N_2) \ . \tag{8}$$

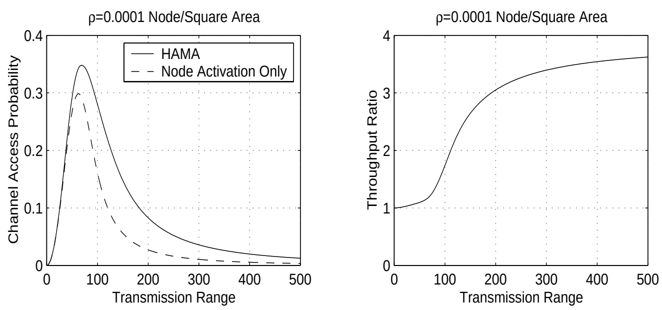

Figure 7. Channel access probability of HAMA and NAMA

The left diagram in Figure 7 compares the advantage of HAMA over the node-activation scheme in terms of channel access probabilities. Because a node barely has any neighbor in a multihop network when the node transmission range is too short, the diagram shows that the system throughput is close to none at around zero transmission range, but it increases quickly to the peak when the transmission range covers around one neighbor on the average. Then network throughput drops when more and more neighbors are contacted and the contention level increases.

The right diagram in Figure 7 shows the performance ratio between HAMA and the node activation scheme. At shorter transmission ranges, HAMA performs very similar to node activation, because nodes are sparsely connected, and node activations are largely BT-kind. When transmission range increases, HAMA obtains more and more opportunities to leverage its unicast capability and the relative throughput also increases more than three times that of the simple node activation.

4.2 Comparison with CSMA and CSMA/CA

The throughput of HAMA is compared with that of idealized CSMA and CSMA/CA protocols, which are analyzed in [18] and [17], where only unicast packets are considered for transmissions.

However, HAMA is modeled differently from CSMA and CSMA/CA. In HAMA, a time slot can carry a complete data packet, while the time slot for CSMA(/CA) only lasts for the duration of a channel round-trip propagation delay, and multiple time slots are used to transmit a data packet once the channel is successfully acquired. In addition, the analyses by [17] [18] assumed a scenario in which a node always has a data packet to utilize the channel, which is not assumed for the throughput analyses of HAMA, because using the heavy-load approximation in HAMA would always result in the maximum network capacity due to the collision-freedom in scheduled channel access schemes.

The probability of channel access at each time slot in CSMA is parameterized by the symbol p'. For comparison purposes, we assume that *every attempt* to access the channel in CSMA or CSMA/CA is an *indication* of a packet arrival at the node. Though the attempt may not succeed eventually for CSMA and CSMA/CA due to packet or RTS/CTS signal collisions in the common channel, and end up dropping the packet, conflict-free scheduling protocols can always deliver the packet if it ever tries to access the channel. We also assume that no packet arrives during the packet transmission. Accordingly, the traffic load for a node is equivalent to the portion of time for transmissions at the node. Denote the average packet size as l_{data}, the traffic load for a node is presented by

$$\lambda = \frac{l_{data}}{1/p' + l_{data}} = \frac{p' l_{data}}{1 + p' l_{data}}$$

because the average interval between successive transmissions follows Geometric distribution with parameter p'.

The network throughput is measured by the successful data packet transmission rate within the one-hop neighborhood of a node in [17] [18], instead of the whole network. Therefore, the comparable network throughput in HAMA is the sum of the packet transmissions by each node and all of its one-hop neighbors. That is:

$$S_{HAMA} = \sum_{k \in N_i^1 \cup \{i\}} \min(\lambda_k, q_k) \ .$$

We reuse the symbol N in this section to represent the number of one-hop neighbors of a node, which is the same as N_1 defined in Section 4.1. Because every node is assigned the same load λ, and has the same channel access probability, q_{HAMA}, given by Eq. (7), the throughput of HAMA becomes

$$S_{HAMA} = N \cdot \min(\lambda,\ q_{HAMA}) \ .$$

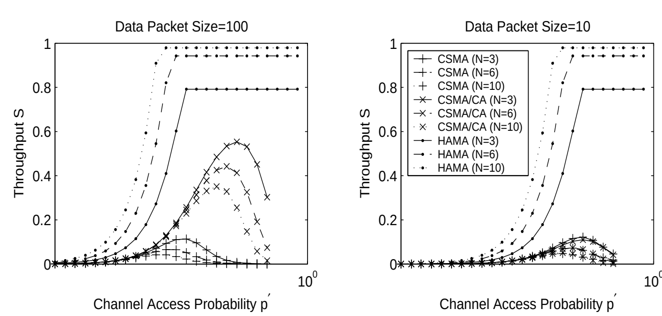

Figure 8. Comparison between HAMA and CSMA, CSMA/CA

Figure 8 compares the throughput attributes of HAMA, and the idealized CSMA [18] and CSMA/CA [17] protocols, in two scenarios. The first scenario assumes that data packets last for $l_{data} = 100$ time slots in CSMA and CSMA/CA, and the second assumes 10-time-slot packet size average.

HAMA provides higher throughput than CSMA and CSMA/CA because of its collision-freedom even when the network is heavily loaded. Whereas HAMA maintains constant throughput in the network under heavy load, the throughput provided by CSMA and CSMA/CA suffers. In contrast to the critical role of packet size in the throughput of CSMA and CSMA/CA, it is almost irrelevant in HAMA, except for shifting the points of reaching the network capacity.

4.3 Comparison with Other Scheduling Protocols

Channel assignment problems in the time, frequency and code domains have traditionally been treated as graph coloring problems. A k-coloring on the graph of a multihop network allows simultaneous conflict-free activations of those entities in the same color, thus achieving efficient temporal and spatial reuse of the available bandwidth. However, the efficiency of the coloring algorithm may suffer from the fact that some of the colors are so rarely used that not enough nodes are assigned to transmit in the corresponding channels.

HAMA is based on a different approach to graph coloring. Because the scheduling is dynamic, and a different schedule is established in each time slot, only two colors are needed for two possible states, transmission or reception. The color for activations is used to the maximal extent in each contention situation.

The delay and throughput attributes of HAMA are studied by comparing it through simulations with scheduling based solely on node activation, called NAMA, and UxDMA [14] in multihop networks with different radio transmission ranges.

In the simulations, we use the normalized *packets per time slot* for both arrival rates and throughput. This metric can be translated into concrete throughput metrics, such as *Mbps* (megabits per second), if the time slot sizes and the channel bandwidth are instantiated.

The simulations were conducted in networks generated by randomly placing 100 nodes within an area of 1000×1000 square meters. To simulate an infinite plane that has constant node placement density, the opposite sides of the square are seamed together, which visually turns the square area into a torus. The power of the transceiver on each node was set to 100, 200, 300, 400 meters, respectively, so that the network topology and contention levels in these simulations vary accordingly. The simulations are guided by the following parameters and behaviors:

- The network topologies remain static during the simulations to examine the performance of the scheduling algorithms only.
- Signal propagation in the channel follows the free-space model and the effective range of the radios is determined by the power level of the radio. Radiation energy outside the effective transmission range of the radio is considered negligible interference to other communication. All radios have the same transmission range.
- Each node has an unlimited buffer for data packets.
- 30 pseudo-noise codes are available for code assignments, i.e., $|C_{pn}| = 30$.
- Packet arrivals are modeled as Poisson arrivals, and packets are served in First-In First-Out (FIFO) order. Only one packet can be transmitted in a time slot.
- All nodes have the same broadcast packet arrival rate for all protocols (HAMA, NAMA and UxDMA). In addition, HAMA is loaded with the same amount of unicast traffic as broadcast traffic to manifest the unicast capability of HAMA. Therefore, the overall load for HAMA is twice as much as that of NAMA and UxDMA. The destinations of the unicast packets in HAMA are evenly distributed over all one-hop neighbors.
- The duration of the simulation is 100,000 time slots, long enough to collect the metrics of interests.

We note that assuming static topologies does not favor HAMA over NAMA or UxDMA, because the same network topologies are used. Nonetheless, exchanging full-topology information, as required by UxDMA, in a dynamic network would be far more challenging that exchanging the identifiers of nodes within two hops of each node.

UxDMA adopts a constraint set that is suitable for broadcast activations as NAMA does, and is give by

$$\text{UxDMA-NAMA} = \{V_{tr}^0, V_{tt}^1\} .$$

The meaning of each symbol is illustrated by Figure 9, where the solid dots are activated nodes (transmitters), and the circle is the receiver. Constraint V_{tr}^0 forbids a node from transmitting and receiving at the same time, while V_{tt}^1 eliminates hidden terminal problem and direct interference.

Figure 10 and 11 show the average packet delay and the *network-wise* throughput of HAMA, NAMA and UxDMA-NAMA. UxDMA-NAMA is better than HAMA and NAMA at broadcasting in some of the multihop networks, owing to

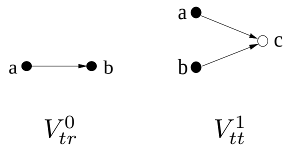

Figure 9. Constraints used by UxDMA for NAMA

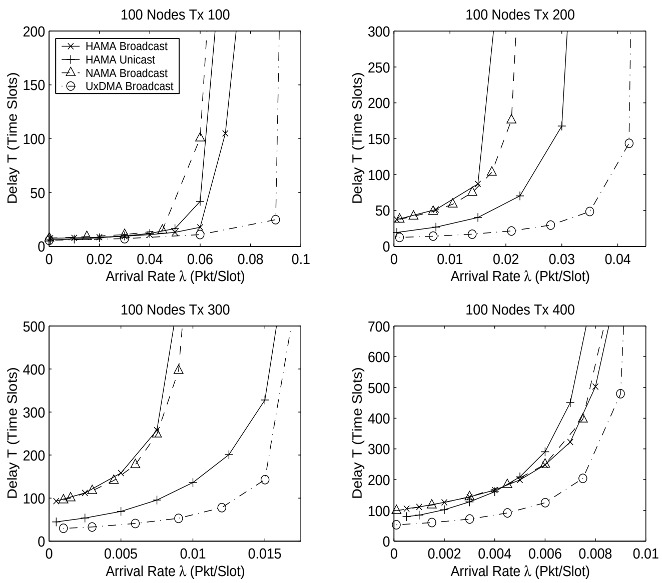

Figure 10. Average packet delays in multihop networks

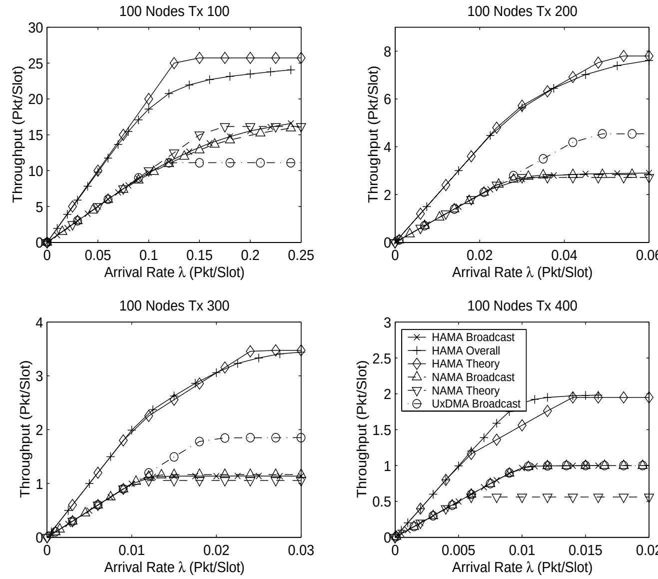

Figure 11. Packet throughput in multihop networks

its global knowledge about the topologies of the networks. However, HAMA outperforms UxDMA-NAMA in network throughput overall.

In Figure 11, analytical throughput lines are also added in each set of multihop network simulations for HAMA and NAMA, respectively. The analytical throughput is denoted by the lines with diamonds for HAMA, and by the lines with top-down triangles for NAMA. The following paragraphs derive the analytical throughput for both HAMA and NAMA:

In the simulations, network nodes are assumed homogeneous in terms of contention level and packet arrival rate. Hence for every node in the network, we denote its unicast packet arrival rate by λ_u, its broadcast arrival rate by λ_b, the probability of channel access for unicast by q_u, and the probability of channel access for broadcast by q_b. The channel access probabilities of a single node during each time slot in HAMA and NAMA are given in Eq. (7) and Eq. (8), respectively. Obviously, the packet throughput per node is the minimum of the channel access probability and the packet arrival rate for transmission.

Let N denote the number of nodes in the network. Because NAMA is only capable of supporting broadcasting, the network throughput for NAMA is:

$$S = N \cdot \min(\lambda_b, q_b).$$

In contrast, HAMA can schedule both unicast and broadcast transmissions, and unicast packet can be sent in a broadcast time slot in the absence of broadcast packets (HAMA lines 48-52). Accordingly, the network throughput for HAMA is represented by:

$$S = \begin{cases} N \cdot (\lambda_b + \min(\lambda_u, q_b - \lambda_b + q_u)), & q_b \geq \lambda_b \\ N \cdot (q_b + \min(\lambda_u, q_u)), & q_b < \lambda_b \end{cases}$$

$$= N \cdot (min(\lambda_b,\ q_b) + \min(\lambda_u,\ \max(q_b - \lambda_b + q_u,\ q_u))).$$

Figure 11 shows that the theoretic analyses follow the simulations closely, except for the deviations of NAMA in the fourth diagram of Figure 11. This is due to the fact that the size of the network area is limited (a $1000 \times 1000m^2$ torus), and that some two-hop neighbors are overlapped at 400-meter transmission such that the two-hop contenders of each node are much fewer than those under the infinite network assumption.

Overall, HAMA achieves much better performance than NAMA with only a little more processing required on the neighbor information. Comparing HAMA with UxDMA, which uses global topology information, HAMA sustains similar broadcasting throughput, in addition to the extra opportunities for sending unicast traffic. The dependence on only two-hop neighbor information is also a big advantage over UxDMA.

5 Conclusion

We have introduced HAMA, a new distributed channel access scheduling protocol that dynamically determines the node- and link-activation schedule for both broadcast and unicast traffic. HAMA is remarkably simple, requires only two-hop neighborhood information, and avoids the complexities of prior conflict-free scheduling approaches that demand global topology information. We have also specified and analyzed a neighbor protocol for coordinating two-hop neighbor information in networks with dynamic topologies. The performance of HAMA was compared by analyses or simulations with that of idealized CSMA and CSMA/CA, dynamic scheduling based on node activation, and UxDMA. The results of our analyses clearly show that HAMA is far more effective than CSMA, CSMA/CA and scheduling based on node activation, and that it renders comparable performance to that of UxDMA without requiring to maintain complete topology information at each node. As such, HAMA constitutes the most effective protocol for conflict-free channel access that does not require complete topology information.

References

[1] L. Bao and J. Garcia-Luna-Aceves. A New Approach to Channel Access Scheduling for Ad Hoc Networks. In *Proc. ACM Seventh Annual International Conference on Mobile Computing and networking*, Rome, Italy, Jul. 16-21 2001.

[2] R. Battiti, A. Bertossi, and M. Brunato. Distributed Code Assignment in Multihop Radio Networks: Object-Oriented Software Simulations. In *Proc. International Conference on Software, Telecommunications and Computer Networks (SoftCOM2000)*, volume 1, pages 157–165, Oct. 10-14 2000.

[3] I. Chlamtac, A. Farago, and H. Zhang. Time-spread multiple-access (TSMA) protocols for multihop mobile radio networks. *IEEE/ACM Transactions on Networking*, 6(5):804–12, Dec. 1997.

[4] I. Cidon and M. Sidi. Distributed assignment algorithms for multihop packet radio networks. *IEEE Transactions on Computers*, 38(10):1353–61, Oct 1989.

[5] B. Crow, I. Widjaja, L. Kim, and P. Sakai. IEEE 802.11 Wireless Local Area Networks. *IEEE Communications Magazine*, 35(9):116–26, Sept 1997.

[6] A. Ephremides and T. Truong. Scheduling broadcasts in multihop radio networks. *IEEE Transactions on Communications*, 38(4):456–60, Apr. 1990.

[7] S. Even, O. Goldreich, S. Moran, and P. Tong. On the NP-completeness of certain network testing problems. *Networks*, 14(1):1–24, Mar. 1984.

[8] L. Hu. Distributed code assignments for CDMA packet radio networks. *IEEE/ACM Transactions on Networking*, 1(6):668–77, Dec. 1993.

[9] M. Joa-Ng and I. Lu. Spread spectrum medium access protocol with collision avoidance in mobile ad-hoc wireless network. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, pages 776–83, New York, NY, USA, Mar. 21-25 1999.

[10] J. Ju and V. Li. An optimal topology-transparent scheduling method in multihop packet radio networks. *IEEE/ACM Transactions on Networking*, 6(3):298–306, Jun. 1998.

[11] L. Kleinrock and F. Tobagi. Packet switching in radio channels. I. Carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, COM-23(12):1400–16, Dec 1975.

[12] R. Krishnan and J. Sterbenz. An Evaluation of the TSMA Protocol as a Control Channel Mechanism in MMWN. Technical report, BBN Technical Memorandum No. 1279, 2000.

[13] T. Makansi. Transmitter-Oriented Code Assignment for Multihop Radio Net-works. *IEEE Transactions on Communications*, 35(12):1379–82, Dec. 1987.

[14] R. Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks*, 5(2):81–94, 1999.

[15] R. Ramaswami and K. Parhi. Distributed scheduling of broadcasts in a radio network. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, volume 2, pages 497–504, Ottawa, Ont., Canada, Apr. 23-27 1989. IEEE Comput. Soc. Press.

[16] T. Shepard. A channel access scheme for large dense packet radio networks. In *ACM SIGCOMM '96 Conference*, pages 219–30, Stanford, CA, USA, Aug. 26-30 1996.

[17] Y. Wang and J. Garcia-Luna-Aceves. Performance of Collision Avoidance Protocols in Single-Channel Ad Hoc Networks. In *Proc. of IEEE International Conference on Network Protocols (ICNP)*, Paris, France, Nov. 12-15 2002.

[18] L. Wu and P. K. Varshney. Performance Analysis of CSMA and BTMA Protocols in Multihop Networks (I), Single Channel Case. *Information Sciences*, 120(1-4):159–177, 1999.

[19] C. Zhu and M. Corson. A five-phase reservation protocol (FPRP) for mobile ad hoc networks. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, volume 1, pages 322–31, San Francisco, CA, USA, Mar. 29-Apr. 2 1998.